United States General Accounting Office

GAO

Report to Congressional Requesters

December 1994

WEATHER FORECASTING

Improvements Needed in Laboratory Software Development Processes

19960124 062

GAO

United States
General Accounting Office
Washington, D.C. 20548

Accounting and Information
Management Division

B-259114

December 14, 1994

The Honorable George E. Brown, Jr.
Chairman
The Honorable Robert S. Walker
Ranking Minority Member
Committee on Science, Space,
and Technology
House of Representatives

The Honorable Ernest F. Hollings
Chairman, Committee on Commerce,
Science, and Transportation
United States Senate

This report responds to your request that we review the software development capabilities of the organizations responsible for developing a critical enhancement to the National Weather Service's (NWS) Advanced Weather Interactive Processing System (AWIPS), known as the AWIPS Forecast Preparation System (AFPS). AWIPS is an information processing system to support forecasters in acquiring, analyzing, and disseminating weather data from various sources. The AFPS enhancement is to automate additional functions currently performed by the forecasters, thereby streamlining the forecast process and improving forecaster productivity. The National Oceanic and Atmospheric Administration (NOAA), of which NWS is a part, is jointly developing AFPS with NWS through the respective laboratories. Their plan is to first define an AFPS software specification through a series of prototype systems and then develop AFPS production-quality software[1] for direct integration into AWIPS.

Our objective was to determine whether the software development processes of NOAA's Forecast Systems Laboratory (FSL) and NWS' Techniques Development Laboratory (TDL) are adequate to support (1) AFPS software prototyping and (2) AFPS production-quality software development. As a rule, the software development processes needed to write production-quality software are much more rigorous, disciplined, and formal than those used for software prototyping. The laboratories' software development processes in question are software requirements management, project planning, quality assurance, configuration

[1] Production-quality software is software that (1) satisfies its specified functional, performance, and operational requirements and thus is ready for day-to-day use and (2) has effective documentation (for example, programmers' manuals, users' manuals, comments on code, testing procedures and results, etc.) to support system operations and maintenance.

management, and tracking and oversight. A detailed explanation of these processes is presented in appendix I.

Results in Brief

The software development processes that FSL and TDL have in place and are following in developing AFPS are adequate to support NWS' near-term AFPS development activities—prototyping for the purpose of defining AFPS requirements and preparing a software specification. However, the laboratories' processes are not adequate to achieve NWS' ultimate objective—developing production-quality AFPS code that NWS can give to the AWIPS contractor for direct integration into AWIPS. Unless the laboratories introduce formality, rigor, and discipline into their software development processes before they begin writing production-quality code, AFPS and AWIPS will likely perform poorly, be delivered late, and cost considerably more than planned. Officials for both laboratories acknowledged their respective process limitations relative to developing production-quality code, and stated that needed improvements are planned.

Background

Since the early 1980s, NWS has been modernizing its observing, information processing, and communications systems to improve the accuracy, timeliness, and efficiency of weather forecasts and warnings. The modernization includes four new major systems—AWIPS, the Next Generation Geostationary Operational Environmental Satellites, Next Generation Weather Radars, and Automated Surface Observing Systems. The Department of Commerce's Deputy Under Secretary for Oceans and Atmosphere is responsible for the modernization.

AWIPS Description and Status

As discussed in our March 1994 report,[2] the heart of the modernization is AWIPS. AWIPS will allow forecasters to integrate, manipulate, and analyze the vast amounts of data that are expected to be available from the new or improved observing and information processing systems.

NOAA awarded the AWIPS contract to the Planning and Research Corporation (PRC) in December 1992 with an atypical arrangement in which NWS would supply PRC with certain specifications in the form of existing and to-be-developed code. Under this arrangement, PRC would then decide, with NOAA's and NWS' approval, whether the code was of

[2] Weather Forecasting: Systems Architecture Needed for National Weather Service Modernization (GAO/AIMD-94-28, March 11, 1994).

sufficient quality to be directly incorporated into AWIPS or whether it required rewriting.

NWS is now in the process of restructuring the AWIPS program and renegotiating the AWIPS contract with PRC. These changes are in response to AWIPS design problems and program delays. Although all the details surrounding program and contract changes have not been defined, NWS officials told us that the restructured program will, among other things, assign responsibility for developing AWIPS application software to the government. This government-developed application software will then be provided to PRC, which will be responsible for providing the AWIPS platform (that is, hardware and systems software environment), integrating the applications with the platforms, and ensuring overall system quality. According to these officials, PRC will be contractually required to use the government-furnished code, including the AFPS code, as delivered, and integrate it into AWIPS (that is, treating it as government-furnished equipment). However, the specific contract terms that define such things as system quality and responsibility for poor system performance have not been specified.

AFPS Description and Status

AFPS is part of the NWS to-be-provided application software. Estimates of its size range from 100,000 to 250,000 lines of code. The system is expected to streamline the forecast process and improve forecaster productivity through automated:

- Forecast visualization - The process of forming an on-screen graphical forecast image based on observations, direct model input, numerical and statistical forecasts, climatology, and other data, as well as experience and training.
- Graphical forecast editing - The process of graphically entering and revising watch, warning, and advisory information used in the original visualized forecast.
- Text generation - The process of automatically producing text based on the graphical representation of the forecast, thus relieving the forecasters of repetitive and time-consuming typing responsibilities.

The laboratories are currently defining and validating AFPS requirements through the use of prototyping techniques. Prototyping is the iterative process of quickly coding, evaluating, and refining less than complete versions of a system. As such, a system prototype is not intended to be an end product or final system; instead, the prototype is a learning tool or a series of learning tools. Prototyping may be undertaken to conduct research (for example, to prove a concept, determine a project's feasibility, or define system requirements), and thus, once the goals are achieved, the prototype can be discarded. However, prototypes can also be retained and used as a foundation for enhancement or be repackaged as a final product.

To date, the laboratories have developed four AFPS iterations. According to laboratory officials, their initial AFPS prototype was a "throw away." The next prototype, however, formed the foundation on which additional system functions and features have and will continue to be added in building increasingly more capable iterations. The laboratories' plan is to continue prototyping until AFPS is refined to the point that it can be placed in an NWS field office for validation by weather forecasters in an operational setting. On the basis of what is learned in the field office, AFPS is to be further refined before providing it to NWS for additional testing and refinement. NWS is then to deliver the code to PRC for integration into AWIPS. According to laboratory, NOAA, and NWS officials, the current plan is to deliver code that requires no rewrite before it is integrated into AWIPS. The officials emphasized that their goal is to develop production-quality code.

Importance of Software Development Discipline

Software development has proven to be the "Achilles heel" of many system development projects. Frequently these projects are delivered late, exceed budgets, and perform poorly because the software development was not guided by disciplined engineering processes.

The rigor and formality of the processes needed to successfully develop software are determined by the desired outcome. If the goal is to develop one or more prototypes that are to be used as learning tools and then discarded, formal software processes and extensive documentation are unnecessary. However, if the desired outcome is production-quality code for the final system, more rigorous and stringent software engineering processes should guide the development effort.[3]

Scope and Methodology

To determine the laboratories' development capabilities, we reviewed key software development documents, including the AFPS development plan, software development and documentation guidelines, summaries of prototyping cycles, and quarterly reports. In addition, we interviewed NOAA

[3] Rigorous software engineering processes are those that are structured, well documented, and systematically implemented and monitored. See appendix II for specific examples of such processes.

and NWS officials at TDL and FSL about their processes. To identify the software development processes that can reasonably be expected to be in place when prototyping a system versus developing production-quality software, we researched relevant literature and government and industry standards, reviewed the Capability Maturity Model developed by Carnegie Mellon University's Software Engineering Institute (SEI), and interviewed SEI officials. SEI's Capability Maturity Model is a tool for assessing organizations' ability to develop software in accordance with modern software engineering methods. This tool focuses on the maturity of certain software development processes. The processes applicable to the laboratories are (1) software requirements management, (2) project planning, (3) quality assurance, (4) configuration management, and (5) tracking and oversight.

We performed our work at NOAA and NWS headquarters and at NWS' TDL in Silver Spring, Maryland; NOAA's FSL in Boulder, Colorado; and SEI in Pittsburgh, Pennsylvania. Our work was performed from May 1994 through October 1994, in accordance with generally accepted government auditing standards.

Laboratories' Processes Are Adequate for AFPS Software Prototyping

The processes that the laboratories have in place for software requirements management, project planning, quality assurance, configuration management, and tracking and oversight are adequate for AFPS prototyping. Laboratory activities in each of the five process areas that we reviewed satisfied the level of structure and documentation advocated by SEI for determining requirements and defining software specifications through system prototyping. In particular, SEI officials stated that prototyping is an investigative process that typically requires flexible processes so as not to overwhelm the developers' creativity. Accordingly, FSL has adopted a process known as the spiral model of software development to define requirements. This process involves iterative builds, each enhancing the previous one. TDL and FSL have also augmented the use of this model with some software development formality, such as the use of documented software coding guidelines and a documented software development plan.

Laboratories' Processes Are Not Adequate for Developing AFPS Production-quality Software

The laboratories do not, however, have in place the software development processes that are needed to develop high-quality, production code. Instead, their processes are largely informal, relying more on the capabilities of laboratory staff rather than clearly defined and documented processes. While certain laboratory process areas (for example, requirements management) are stronger than others (for example, software quality assurance), none possess the full complement of activities that the SEI Capability Maturity Model advocates. Specific examples of where the laboratories are deficient in each process area are described below. Additional examples are provided in appendix II.

- Requirements management: Organizations developing production-quality software should have a software engineering group that reviews and agrees to requirements before code is written to incorporate these requirements into the software. Neither TDL nor FSL have software engineering groups.
- Project planning: The software development plan is the culmination of software project planning. To the laboratories' credit, they have an AFPS software development plan that defines the project's purpose, goals, organizational development responsibilities, software development methodology, and software development schedules. However, their plan lacks other important elements, including software verification and validation provisions and software metrics.
- Quality assurance: The software quality assurance plan is the centerpiece of an effective quality assurance program. This plan defines the activities necessary to ensure that software development processes and products conform to applicable requirements and standards. In addition, an independent software quality assurance group should review and audit the software engineering activities to ensure compliance. The laboratories do not have a software quality assurance plan or group.
- Configuration management: A software configuration management plan should exist that clearly defines the procedures for identifying, accounting for, and reporting on changes to software items that are under configuration control. Neither laboratory has a software configuration management plan.
- Tracking and oversight: To ensure that a software development project is proceeding as planned, formal reviews to communicate accomplishments and results of the project software engineering should be conducted at selected project milestones. The laboratories do not conduct such reviews.

FSL and TDL officials agreed that their software development processes are not adequate for production-quality development activities. They stated that actions are planned to improve the maturity of these processes. For example, FSL plans to hire an individual to establish and manage an independent quality assurance program, including development and implementation of a quality assurance plan. Also, FSL plans to improve its requirements management process by (1) documenting the process, (2) developing a technique for mapping system requirements to the system and software designs, and (3) documenting changes to the requirements baseline.

Conclusions

Although FSL and TDL have adequate software development processes for defining AFPS requirements via prototyping, neither has adequate processes for developing production-quality code, which is NWS' ultimate goal on AFPS. Without these processes, NWS is exposing AFPS, as well as AWIPS, to unnecessary cost, schedule, and performance risks.

Recommendations

In light of NWS' plan to provide the AWIPS contractor with production-quality software for direct integration into AWIPS, we recommend that the Secretary of Commerce direct the Deputy Under Secretary for Oceans and Atmosphere to have FSL and TDL strengthen their software development processes for requirements management, project planning, quality assurance, configuration management, and tracking and oversight before beginning development of any production-quality code. The crucial processes are outlined in appendix II of this report.

We discussed the contents of this report with TDL and FSL officials, the AWIPS Technical Director, and the NWS Modernization Systems Manager. These officials generally agreed with the information presented. We have incorporated their comments where appropriate.

We are sending copies of this report to the Secretary of Commerce, the Director of the Office of Management and Budget, and interested congressional committees. Copies will also be made available to others upon request.

Please call me at (202) 512-6253 if you or your staff have any questions concerning this report. Other major contributors are listed in appendix III.

Joel Willemssen

Joel C. Willemssen
Director, Information Resources
Management/Resources, Community,
and Economic Development

Contents

Abbreviations

AFPS	AWIPS Forecast Preparation System
AWIPS	Advanced Weather Interactive Processing System
FSL	Forecast Systems Laboratory
NOAA	National Oceanic and Atmospheric Administration
NWS	National Weather Service
PRC	Planning and Research Corporation
SCM	software configuration management
SEI	Software Engineering Institute
SQA	software quality assurance
TDL	Techniques Development Laboratory

Appendix I

Software Development Process Areas

Table I.1 provides the software development process areas we evaluated and their definitions.

Table I.1: Software Development Process Areas

Software requirements management	This process involves establishing and maintaining an agreement with the customer for both the technical and nontechnical requirements for the software project. This agreement forms the basis for estimating, planning, performing, and tracking the software project's activities throughout the software life cycle.
Software project planning	This process is a subset of the overall project planning. It involves defining each major software project task, estimating the time and resources required to accomplish it, and tracking and controlling actual software production against these and other production goals. The centerpiece of software project planning is the software development plan.
Software quality assurance	This process is the planned and systematic set of actions necessary to provide sufficient confidence that the software product conforms to established requirements. Effective quality assurance should be (1) managed by an organization independent of the organizations managing the project and developing the software products, (2) managed by an organization that has the authority to establish and enforce software quality standards and procedures, (3) based on predetermined software metrics, and (4) managed by documented processes that are shared among parties participating in the project.
Software configuration management	This process is the means by which changes to software products are controlled. It includes identification of software products to be controlled, accounting for changes to these controlled products, and reporting on the status of these products.
Software tracking and oversight	This process provides insight into project progress and provides a basis for reporting on project status. It is accomplished by measuring ongoing software development activities and comparing the measurements against documented estimates, commitments, plans, and industry norms.

Comparison of Crucial Production Software Development Activities With Those at TDL and FSL

This appendix identifies the activities[1] that are expected of leading software development organizations and contrasts these with the activities currently performed at TDL and FSL. While this is not a comprehensive list of activities, it highlights the most crucial activities desired in each of the software development process areas.

Table II.1: Software Requirements Management Process

Crucial activities	TDL and FSL activities
The requirements are documented in a consistent format and are clearly stated, verifiable, and testable.	The requirements are documented with each prototype iteration. Each iteration more clearly defines the baseline requirements.
The software engineering group reviews and agrees to the requirements before they are incorporated into the software efforts.	A formal software engineering group does not exist.
The requirements form the basis for the software plans, products, and activities.	The AFPS requirements are still being developed, and thus such documents and activities do not yet exist.
Changes to the requirements are appropriately reviewed and incorporated into the software efforts.	Requirements changes are reviewed by a technical coordinator and are incorporated into the next prototype iteration.

[1] The activities were derived from SEI's Capability Maturity Model.

Table II.2 Software Project Planning Process

Crucial activities	TDL and FSL activities
The software engineering group is an active participant in proposing and planning the project throughout its life.	There is no formal software engineering group, but both TDL and FSL programmers participate in the project's planning activities.
Software planning is initiated in the early stages of, and in parallel with, overall project planning.	The AFPS software development plan was developed in conjunction with the AFPS concept document.
Software planning data are recorded for use by the project.	This activity does not exist.
Senior management reviews and approves all commitments made to individuals and groups external to the organization.	No arrangements with external organizations exist.
A software life-cycle model with predetermined stages of manageable size is identified or defined.	A software life-cycle model does not exist.
The project's software development plan is developed according to a documented procedure and addresses all software activities.	The AFPS software development plan is documented in the AFPS concept document and addresses most software activities.
Software products and software process specifications that are needed to establish and maintain stability of the software activities are explicitly identified as controlled project baseline items.	Software product and process specifications (for example, design documents, programming guidelines) exist but are not identified as controlled project baseline items.
Estimates for the size of the software products, the software development resources and costs, the critical target computer resources, and the software schedule are derived according to a documented procedure.	A documented procedure to derive these items does not exist.
The software technical, cost, resource, and schedule risks are assessed and documented.	Such risks are identified and assessed informally; however, they are not documented.

Table II.3: Software Quality Assurance Process

Crucial activities	TDL and FSL activities
A software quality assurance (SQA) plan is prepared for each software project according to a documented procedure, and SQA activities are performed in accordance with the SQA plan.	An SQA plan does not exist.
The SQA group participates in the preparation, review, and approval of the project's software development plan, process specifications, standards, and procedures.	An SQA group does not exist.
The SQA group reviews and audits the software engineering activities to ensure process compliance.	The FSL Technical Coordinator reviews code and conducts code walk-throughs. TDL staff have peer code reviews and walk-throughs. However, a separate SQA group does not exist.
The SQA group reviews representative samples of deliverable and designated nondeliverable software products to ensure compliance with the designated process requirements.	Such activities do not occur.
The SQA group regularly reports its reviews and audits to the software engineering staff and managers.	Such activities do not occur.
Deviations identified in the software engineering activities are documented and handled according to a documented procedure.	Deviations identified as a result of the FSL Technical Coordinator and TDL peer reviews are addressed but not documented nor handled according to documented procedures.

Table II.4: Software Configuration Management Process

Crucial activities	TDL and FSL activities
A documented software configuration management (SCM) plan exists and is used as the basis for performing SCM activities.	An SCM plan does not exist.
A configuration management library system is established as a repository for the software baselines.	A configuration management library system for managing the software baseline does not exist. However, the laboratories use documented, automated tools for configuration control of the software products.
The software engineering products and process specifications to be placed under configuration management are identified.	Such products and processes have not been identified for placement under configuration control.
A documented procedure is followed for initiating, recording, reviewing, approving, and tracking change requests and trouble reports for all configuration items.	A documented procedure does not exist for tracking and controlling change requests and trouble reports.
A documented procedure is followed to create and control the release of software baseline products.	A documented procedure does not exist to control software baselines.
A documented procedure is followed to record the status of configuration items and change requests, and to control changes to configuration items.	Such documented procedures do not exist.

Table II.5: Software Tracking and Oversight Process

Crucial activities	TDL and FSL activities
A documented software development plan is used for tracking the software activities and communicating status.	A software development plan is not used for tracking the software activities; however, such activities are tracked and communicated through quarterly reports.
Senior management reviews and approves all commitments to individuals and groups external to the organization.	No arrangements with external organizations exist.
Approved changes to software commitments or commitments affecting the software activities are explicitly communicated to the staff and managers of the software engineering group.	Software changes are communicated informally between laboratory officials.
The project's size, costs, critical target computer resources, software schedule, and software engineering activities are tracked and corrective actions are taken.	These activities are not tracked by any formal means.
The software technical, cost, resource, and schedule risks are tracked throughout the life of the project.	These risks are not tracked by any formal means.
Actual measured data and replanning data for the software project tracking activities are recorded for use by the software engineering staff and managers.	Such activities do not occur.
The software engineering staff and managers conduct regular reviews to track technical progress, plans, performance, and issues against software development plans.	Such regular reviews to track such items against software development plans do not occur.
Formal reviews to address the accomplishments and results of project software engineering are conducted at selected project milestones.	Formal reviews are not conducted.

Major Contributors to This Report

Accounting and Information Management Division, Washington, D.C.

Dr. Rona B. Stillman, Chief Scientist for Computers and Communications
Randolph C. Hite, Assistant Director
Keith A. Rhodes, Technical Assistant Director
David A. Powner, Evaluator-in-Charge
Colleen M. Phillips, Senior Evaluator

Ordering Information

The first copy of each GAO report and testimony is free. Additional copies are $2 each. Orders should be sent to the following address, accompanied by a check or money order made out to the Superintendent of Documents, when necessary. Orders for 100 or more copies to be mailed to a single address are discounted 25 percent.

Orders by mail:

U.S. General Accounting Office
P.O. Box 6015
Gaithersburg, MD 20884-6015

or visit:

Room 1100
700 4th St. NW (corner of 4th and G Sts. NW)
U.S. General Accounting Office
Washington, DC

Orders may also be placed by calling (202) 512-6000 or by using fax number (301) 258-4066, or TDD (301) 413-0006.

Each day, GAO issues a list of newly available reports and testimony. To receive facsimile copies of the daily list or any list from the past 30 days, please call (301) 258-4097 using a touchtone phone. A recorded menu will provide information on how to obtain these lists.

PRINTED ON RECYCLED PAPER